

## II-126 工業用ロボットとの対話に関する ソフトウェアシステムについて

向 殿 政 男

### On a Conversational Soft Ware System of Industrial Robot

Masao MUKAIDONO

#### Abstract

Studies of operating a manipulator with a digital computer have been mainly pursued in order to apply a remote manipulator for the exploration of both outer space and the ocean depths. On the other hand, an idea of applying a manipulator with some degree of intelligence for industries has also been positively adopted for some years. The latter application is already being realized as the so-called industrial robot. Versatran and Unimate are representative of this type. As industrial robots of this sort, however, have only simple functions such as recording and rerecording, their applications seem very limited.

In this paper, a control system of an industrial robot with conversations between an operator and a computer is described. This control system is similar to MANTRAN developed for supervisory control of a remote manipulator at M. I. T., and has the following features which are suitable for applications in industrial fields :

- 1) by keeping watch on a robot, an operator can teach the robot many kinds of tasks by means of coversations with a computer;
- 2) a robot becomes more and more informed by accumulating experiences, and the operator's burden gradually becomes lighter ;
- 3) in the case that a robot cannot make a decision, an operator is ready to make a suitable decision for it ;
- 4) it is not necessary to prepare programs for every kind of situation at first since a robot can construct new working instructions freely and easily by combining working programs already accumulated.

#### § 1. まえがき

ロボットの研究が、単調な繰り返し作業や危険で人間が直接手を触れることができない作業を行なう工業用マニピュレータの次に来たるべきものとして、最近注目を浴びている。しかし、工業用マニピュレータと、智能を有するロボット——人工知能ロボット——との間には質的に異なったかなりのギャップがあるが、最近、急激に発展している電子計算機技術と工業用マニ

ピュレータとを結合することによりある程度の判断機能を持った、広範囲な用途に使用できる、万能で融通性のある工業用マニピュレータの研究が、ロボット研究の第一段階として、現在試みられつつある。

最初にマニピュレータを電子計算機を用いて動かすことを試みたのは、恐らく、H. A. Ernst<sup>\*1</sup>であろう。いままでの単能な自動機械に磁気テープと簡単な記憶装置とを組み合わせ、プログラム制御することにより融通性を持たせた工業用ロボット (Versatran, Unimat-

e) は、現在、実用に供せられている。しかし、これらは融通性にも限度があり、作業内容も比較的簡単なルーチンワークに限られている。一方、月などの非常な遠隔地に置かれた工業用マニピュレータを地上より制御するときの時間遅れの問題を解決するために、マニピュレータと小形計算機とを直結して、この小形計算機に地上からおおまかな指令を与えて、細かいことはすべてこの直結した小形計算機がマニピュレータに指令を出して動かすような管理制御システムを、T. B. Sheridan<sup>\*2</sup>や W. R. Ferrell<sup>\*3</sup>らが提唱し、いくつかの実験を試みた。このシステムでは、マニピュレータに作業を指令する人間と計算機との対話が基本となっており、この考えに沿って、D. J. Barber<sup>\*4</sup>は管理制御用計算機言語“MANTRAN”を開発している。

本研究は、工業用ロボットを対象とし、極めて融通性のある高度な計算機制御システムを開発することを目的としている。すなわち、オペレータと計算機との対話によって各種の作業をロボットに教えて記憶させ、今迄記憶している作業を自由に組み合わせて次ぎ次ぎに新しい作業を作り上げ、経験豊かなロボットに成長させていくためのソフトウェアシステムを作り、各種の

実験によっていろいろな問題点を洗い出し、さらに、高度な知能を有してオペレータの負担を軽減するような総合的な工業用ロボット制御システムの可能性を追究している。

## § 2. システムの概要

### 2-1. ハードウェアシステム——工業用ロボットとインターフェース——

実験システム全体の構成を図1に示す。実験室は計算機室とロボット室とからなり、オペレータはテレビカメラを通して計算機室からロボットの動きを見ながら実験できるようになっている。ロボットの制御にはNEAC-3100 (32k語, 18ビット/語, サイクルタイム2 $\mu$ s)を使用した。計算機とロボットとの情報の交換は、OCRアダプタ<sup>\*\*1</sup>を通してそれ専用に製作されたインターフェースによるか、またはAD-DA変換器によって行なわれる。また、オペレータと計算機との対話は入出力用タイプライタにより行なわれる。

工業用ロボットは、A. M. F社のVersatranに模して作られた油研式工業用ロボットを使用した(図2参照)。これは、油圧により駆動し、腕、手首、指先よりなっている。腕は図3のような円筒座標(R,

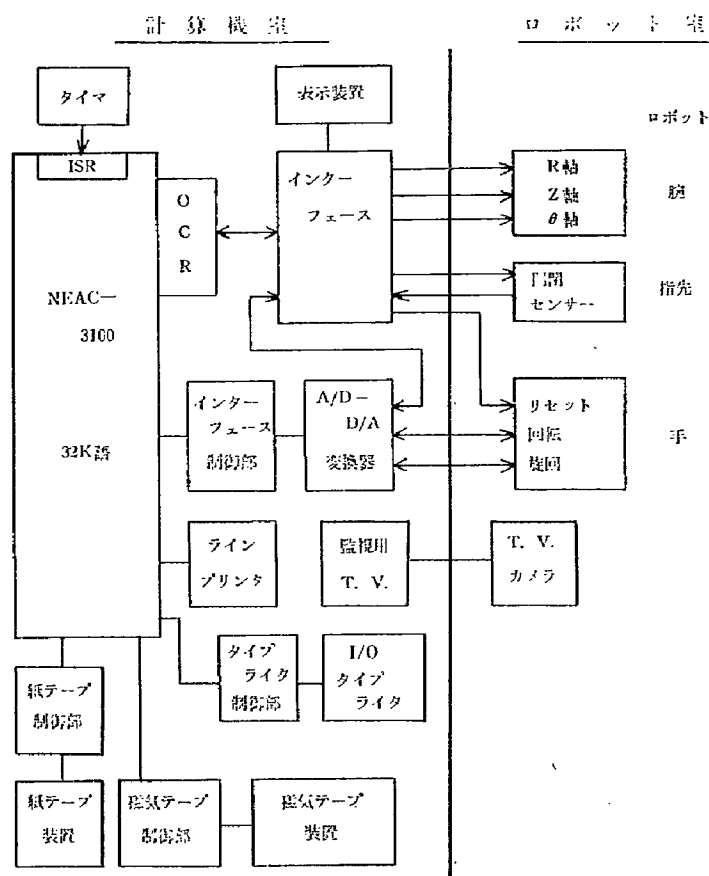


図1 実験システムの構成図

$\theta$ , Z) 上を動かことができ、各軸は、インターフェースを通して計算機より出されるパルスにより制御される。各軸のパルス当りの動作量を表1に示す。手首は、旋回、回転、指先は開閉の各動作ができる。手首

腕	最大範囲	最大速度	動作量	手	最大範囲
			パルス		
Z	800mm	700mm/S	0.69mm	旋回	$\pm 90^\circ$
R	800mm	100mm/S	0.93mm	回転	$\pm 180^\circ$
$\theta$	$240^\circ$	$90^\circ/S$	$0.1^\circ$	開閉	$60^\circ$

表1 ロボットの動作仕様

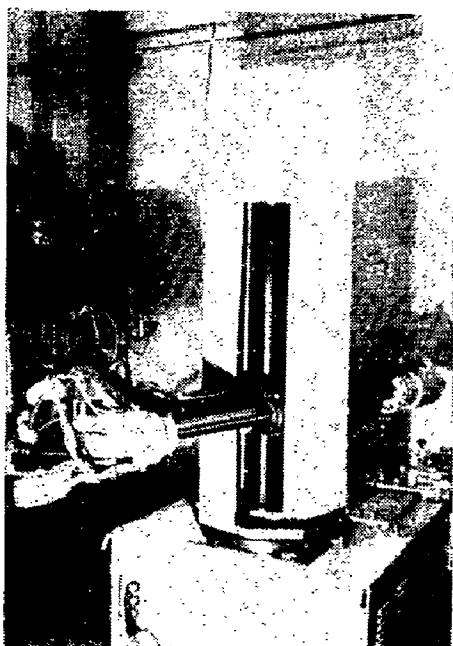


図2 油研式工業用ロボット

は、D-A変換器を通して直接電圧レベルを設定することにより制御される。逆に、A-D変換器により、手首の現在の状態を読み込むこともできる。なお、指先には、マイクロスイッチによるセンサーが設けられており、これらの状態はインターフェースを通じて計算機内に読み込まれる。計算機には、ISR (Interrupt Sense Register) を通して TIMER が接続されており、これにより計算機内での時間測定が行なわれる。インタフェースの機能は、以上の腕の制御パルスの伝達、指先のセンサーによる情報の読み込みの他に、リセット命令の解釈、指令、および不正命令のチェックなどである。また、制御パルスをカウントすることにより腕の現在地を示す Indicator もそなえつけられている。

本論文は、ソフトウェアシステムについて述べることを主眼とするので、ここでは、ハードウェアについては以上の概説にとどめておく。詳しいことは他の論文<sup>5,6</sup>を参照されたい。

## 2-2. ソフトウェアシステム—基本ルーチンと準基本ルーチン—

工業用ロボットが行なえる基本的動作は、ほんの数種類に限られている。どんな複雑な作業でも、最終的にはこの基本動作のシーケンスに分解される訳である。よって、ソフトウェアシステムでロボットを動かすために使用する基本的なサブルーチンも以下に述べるように数少ない基本ルーチンと、この基本ルーチンの特

\*\*1: OCR (Optical Character Reader) アダプタは、光学的文字読取り用に開発、製作されたものであるが、外部装置とのデータのやり取りは、このアダプタを利用して行なっている。6[μS] 当り 6 ビット並列に転送できる。

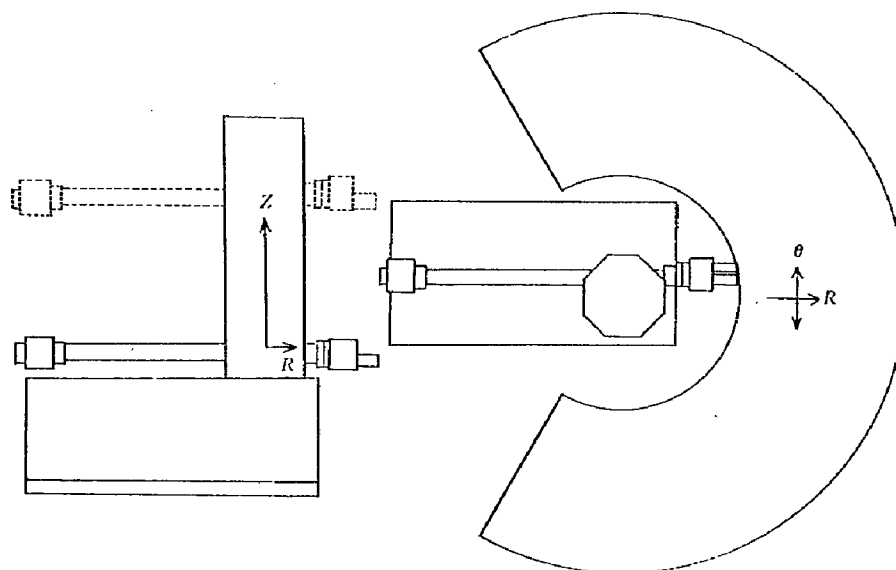


図3 ロボットの動作範囲

殊なものないしはそれらの簡単な組み合わせでできて各種作業に多く用いられる準基本ルーチンとからなっている。すべてのロボットの動作、作業はこれらの基本サブルーチンの組み合わせで実現される。なお、ソフトウェアシステムのすべてのプログラムは、アセンブラと FORTRAN とを混せて書かれており、アセンブラが主体である。

#### A) 基本ルーチン

##### \* GOTO1\*\*2

腕を目的の地点まで移動させるためのプログラムで、絶対座標の点にも移動できるし、相対的にも移動できる。現在は、R,  $\theta$ , Z の順で3軸独立に動かしている。パラメータは、R, Z は mm 単位で、 $\theta$  軸は度単位で与える。

##### \* GRIP

物をつかむために手首を旋回させたり回転させるためのプログラムで、ロボットの腕に固定された座標より見た旋回と回転の角度をパラメータとして、度単位で与える。

##### \* OPEN

指先を開く。

##### \* CLOSE

指先を閉じる。

##### \* GRASP\*\*3

ある指定された範囲内で物体を探索し、物体があったらセンサーの状態で適当にその状況を判断して物体を把握するプログラムである。

##### \* TRANS

ロボットの座標と作業空間に固定された座標との変換、および、直角座標と円筒座標との変換用プログラム。

##### \* P. SET

空間の一点に名前をつけるためのプログラム。

#### B) 準基本ルーチン

##### \* LIFT

腕を現在の位置より指定された値だけ相対的に上げる。

##### \* DROP

腕を現在の位置より指定された値だけ相対的に下げる。

##### \* RESET 1

ロボットに設けられているリセット機能により、腕

をリセットする。

##### \* RESET 2

手首、指先をリセットする。

##### \* APPROC\*\*3

物体が存在するだいたいの位置が与えられたとき、その物体のある指定された近傍まで近づくプログラム。

以上述べた基本ルーチン、準基本ルーチンを基本単位として、タイプライタを通してロボットと対話しながら各種の作業を教えて憶させ、今迄記憶している作業を自由に組み合わせて新しい作業を作り上げていく。このとき、いくつかのパラメータを持った作業も構成できる。また、任意の空間の一点に名前をつけることができるから、以後はその名前でその位置を指定することができる。次の章 (§ 3) で実施例によって、このソフトウェアシステムの特長や使用方法を述べ、§ 4 でその詳しい構造について述べる。

### § 3. ロボットの計算機制御実施例

このソフトウェアシステムを、実施例に従って説明して行く。先ず初めに点の登録について、次に作業（命令）の登録と実行について述べるが、例 1 から例 13 までは、一つの連続した実施例である。以下の例で、※の印で始まる行は、すべてロボット側からの質問ないしは返答である。

例 1) 腕を動かしながら点に名前をつける。

まず、作業を始める場合は最初にロボットの方から

※ナニ オ シマスカ

P. SET

※ウゴクカ (U), ナマエダケカ (N)

U

※RTZ, ナークミアワセテ イレヨ

R+

※SW-OFF. ツヅケルナラ GO ツギナラ NEXT  
GO

※SW-OFF. ツヅケルナラ GO ツギナラ NEXT  
NEXT

※テン ノ ナマエ オ イレテ クダサイ  
POINT 1

※※ POINT 1 オ セット シタ

\*\*2: 腕を直線に添って動かしたり、障害物を探索しながら目的地まで動かせるプログラムなど必要で、これらを GOTO 2, GOTO 3, …… と名づけて、これから開発して行く予定である。

\*\*3: 本論文で述べるすべてのプログラムは、最終的には眼で物を認識するプログラムと結合されて、より高度の智能を有するソフトウェアシステムとする。このとき、眼からだいたいの物体の位置を与えられて、その付近に近づくプログラムが APPROC である。

※ナニ オ シマスカ と質問してくる。P. SET とタイプしてやると、この命令は、空間の点に名前をつけ、これを登録する場合に用いるもので、例に示すようにロボット側から再び質問がくる。Uを入力すれば、腕を実際に動かして、必要などころに来たら止めて、その点に名前をセットできる。Nを入力すれば座標を入力することによって点をセットすることができる。この例ではUを入力している。R+とは、R軸の正の方向に腕を動かすことを意味しており、これによりロボットの腕は動き出す。希望したところまで腕が伸びたら、センススイッチをONすると腕は止まって例に示すように返事をしてくる。センススイッチをOFFにもどしてGOと入力すれば続いて腕はR軸の方向に伸びるし、NEXTと入力すれば点の名前を要求してくる。この例では、今、指先のある空間の地点に POINT 1 という名前をつけている。名前を入力するとこの点の登録は済んで、次の点の登録の操作がくり返される。この例をもう少し続けてみる。

例2) 座標を入力することにより点をセットする。

```
※ウゴクカ(U), ナマエダケカ(N)
N
※テン ノ ナマエ オ イレテ クダサイ
POINT 2
※データ オ イレヨ
※X ハ
-800
※Y ハ
900
※Z ハ
700
※※POINT 2 オ セット シタ
```

例のようにNと答えると、登録すべき点の名前とその座標を要求してくる。X, Y, Z座標の単位はmmである。次の例で、今までに登録した名前を入れて見よう。

例3) 登録済の名前を入力する。ロボットの動けない範囲を入力する。

```
※ウゴクカ(U), ナマエダケカ(N)
N
※テン ノ ナマエ オ イレテ クダサイ
POINT 2
※ POINT 2 ハ アリマス ホカ ノ ナマエ
オ ドウゾ
POINT 3
※データ オ イレヨ
```

```
※X ハ
800
※Y ハ
800
※Z ハ
1900
※※コノ データ デハ ロボット ハ ウゴケナ
イ
※データ オ イレヨ
※X ハ
800
※Y ハ
800
※Z ハ
900
※※ POINT 3 オ セット シタ
※ウゴクカ(U), ナマエダケカ(N)
N
※テン ノ ナマエ オ イレテ クダサイ
NO
※※ P. SET オワリ
```

点の名前を入力する時には、いつでもその点が登録済みであるかどうかをチェックしている。また、腕が動き得る範囲は決まっているから、その範囲以外の点を指定すると例に示すように動けないと返答してくる。点の名前の代りにNOと入力すると、このP. SET命令は終りになり、次の命令を要求してくる。

例4) 今までにセットした点をパラメータとしてGOTOを命令行なう。

```
※ナニ オ シマスカ
GOTO 1 POINT 2
※※ GOTO 1 オワリ
```

これで、前に登録した点、POINT 2に腕は動いたわけである。なお、点の登録は、P. SET命令だけでなく、次のようにしてもできる。

例) 今までに登録されていない点を使う

```
※ナニ オ シマスカ
GOTO 1 POINT 4
※ POINT 4 トイウ テン ハ ナイ セット
スルカ
YES
※データ オ イレヨ
※X ハ
```

-756

※Y ハ

900

※Z ハ

1000

※※ GOTO 1 オワリ

このように、まだ登録されていない点をパラメータとして使用するとそのデータを要求してくる。この例によって、POINT 4 は登録されて、腕は POINT 4 に動いた。

#### 例6) 基本命令の実行

※ナニ オ シマスカ

GOTO 1

※データ オ イレヨ

※X ハ

1000

※Y ハ

500

※Z ハ

700

※テン オ セット スラナラ ナマエオ ドウゾ  
POINT 5

※※ POINT 5 オ セット シタ

※※ GOTO 1 オワリ

※ナニ オ シマスカ

LIFT 90

※テン オ セット スラナラ ナマエ オ ドウ  
ゾ

POINT 6

※※ POINT 6 オ セット シタ

※※ LIFT オワリ

※ナニ オ シマスカ

APPROC 10

※※ APPROC オワリ

※ナニ オ シマスカ

OPEN

※※ OPEN オワリ

※ナニ オ シマスカ

LIFT

※データ ハ

100

※テン オ セット スラナラ ナマエ オ ドウ  
ゾ

NO

※※ LIFT オワリ

これらの例からもわかるように、まず命令の名前を入力し、次にそれが必要とするパラメータを入力すればよい。パラメータの必要な命令にパラメータを与えないと、ロボットの方から必要なデータを要求してくるよう、柔軟性を持たせて作られている。

今迄は、基本命令のみを指令したが、以後、これらの基本命令を組み合わせる新しい命令（作業）を作る方法を示す。

#### 例7) 簡単な作業の構成

※ナニ オ シマスカ

AKERO

※AKERO ハ ワカラナイ セツメイ セヨ  
HIRAKE

※ HIRAKE ハ ワカラナイ セツメイセヨ  
アケロ

※アケロ ハ ワカラナイ セツメイセヨ

OPEN/※ アケロ ハ ワカッタ ツツケヨ

/※ HIRAKE ハ ワカッタ ツツケヨ

/※ AKERO ハ ワカッタ ジツコウシマスカ  
YES

※※ AKERO オワリ

この例に示すように、まだロボットが知らない命令を指令すると、その説明を求めてくる。その新しい命令を説明している途中で再びまだ知らない命令が出てきてもよく、このときにもまたその命令の説明を求めて来て、これは何重になってもよい。／は命令（作業）の説明の完了時にオペレータが入力するものである。この例によって、アケロ、HIRAKE、AKER-O は自動的に登録されて、OPEN とまったく同じ動作を行なうようになる。

#### 例) 作業の構成

※ナニ オ シマスカ

ハコイレ

※ ハコイレ ハ ワカナイ セツメイセヨ

アケロ

APPROC 10

GRASP

LIFT 200

GOTO 1 POINT 1

アケロ

GOTO 1 GENTEN /※ ハコイレハ ワカッタ

```

| ジツコウ シマスカ
| YES
| ※※ ハコイレ オワリ

```

この例題で、GENTEN という点は、座標0の点で、特殊な点として始めからロボットは知っている。この例題によって、目から情報をもらって、目的の物体の約10mm前まで近ずき、センサーを使って探査して物を把握し、20cm持ち上げて POINT 1 に持って行き、はなして原点に帰るという作業ハコイレが構成できたわけである。POINT 1 を箱の上の点とすると箱に物を入れる仕事ということになる。

次に、POINT 1 の部分を可変にしみる。

例9) パラメータを持った作業の構成

```

※ナニ オ シマスカ
HKOIN
※ HKOIN ハ ワカラナイ セツメイセヨ
AKERO
APPROC 10
GRASP
LIFT 200
GOTO 1
AKERO
GOTO 1 GENTEN/※ HKOIN ハ ワカツ
タ ジツコウ シマスカ
NO
※ナニ オ シマスカ
HKOIN POINT 5
※※ HKOIN オワリ

```

前の例と違うところは、登録するときに GOTO 1 にパラメータを入力しなかったことである。このようにして、パラメータを持つ作業を構成することができる。すなわち、上の例によって、HKOIN は、つかんだ物体を離す位置をパラメータとして持つ作業として登録される。このようなパラメータは、現在9個まで設けることができる。次に、多くのパラメータを持った作業を構成する例を示す。

例10) 多くのパラメータを持つ作業の構成

```

※ナニ オ シマスカ
GOGOGO
※ GOGOGO ハ ワカラナイ セツメイセヨ
GOTO 1
GOTO 1

```

```

GOTO 1
GOTO 1
GOTO 1
GOTO 1
GOTO 1 GENTEN/※ GOGOGO ハ ワカ
ツタ ジツコウ シマスカ
NO
※ナニ オ シマスカ
GOGOGO POINT 1 POINT 2 POINT 3
POINT 4 POINT 5 POINT 6
※※ GOGOGO オワリ

```

これにより、パラメータとして与えられた6点を通り、最後に原点に帰る運動が実現できたことになる。なお、このように構成された作業の実行を指令するとき、パラメータを与えないと基本命令の場合と同様に、次の例に示すようにパラメータを要求してくる。

例11) パラメータの必要な作業にパラメータを与えないで指令した時

```

※ナニ オ シマスカ
GOGOGO
※テン ノ ナマエ ハ
POINT 1
※テン ノ ナマエ ハ
POINT 4
※テン ノ ナマエ ハ
POINT 6
※テン ノ ナマエ ハ
POINT 2
※テン ノ ナマエ ハ
GENTEN
※ テン ノ ナマエ ハ
POINT 5
※※ GOGOGO オワリ

```

上の GOGOGO は、GOTO 1 命令のみから構成されていたが、LIFT, DROP, その他パラメータを必要とする登録ずみの作業を含んでいても、同様である。また、途中でパラメータとして登録されていない点を与えると、データを要求してくるのは例5の場合と同様である。

例12) 同じ作業を繰り返す。

```

※ナニ オ シマスカ
HKOIN 4

```

```

** HKOIN オワリ
**ナニ オ シマスカ
GOGOGO POINT 6 POINT 4 POINT 2
POINT 1 POINT 3 POINT 5 #
**SW 2-OFF
** GOGOGO オワリ

```

ある作業を繰り返したい場合は、作業名とパラメータを与えた後に、回数を数字で入力すればよい。上の例では HKOIN を4回繰り返している。数字の代りに#を入力すると何回でも繰り返す。これを止めるためには、センススイッチをONにすればよい。その作業が終わった時点で\*\*SW2-OFF と出力されてその命令が終るから、センススイッチを OFF にして次の作

業に移ればよい。

例I3) すべての作業の終了

```

**ナニ オ シマスカ
STOP
**STOP

```

これにより今まで登録された点や作業は、すべて磁気テープにしまわれて、これらの点や作業はいつでも使うことができる。

以上のようにして、対話をしながら経験豊かなロボットに成長させて行くことができる。

#### § 4. 対話形式ソフトウェア

前章の例題で示した、工業用ロボットと対話をしな

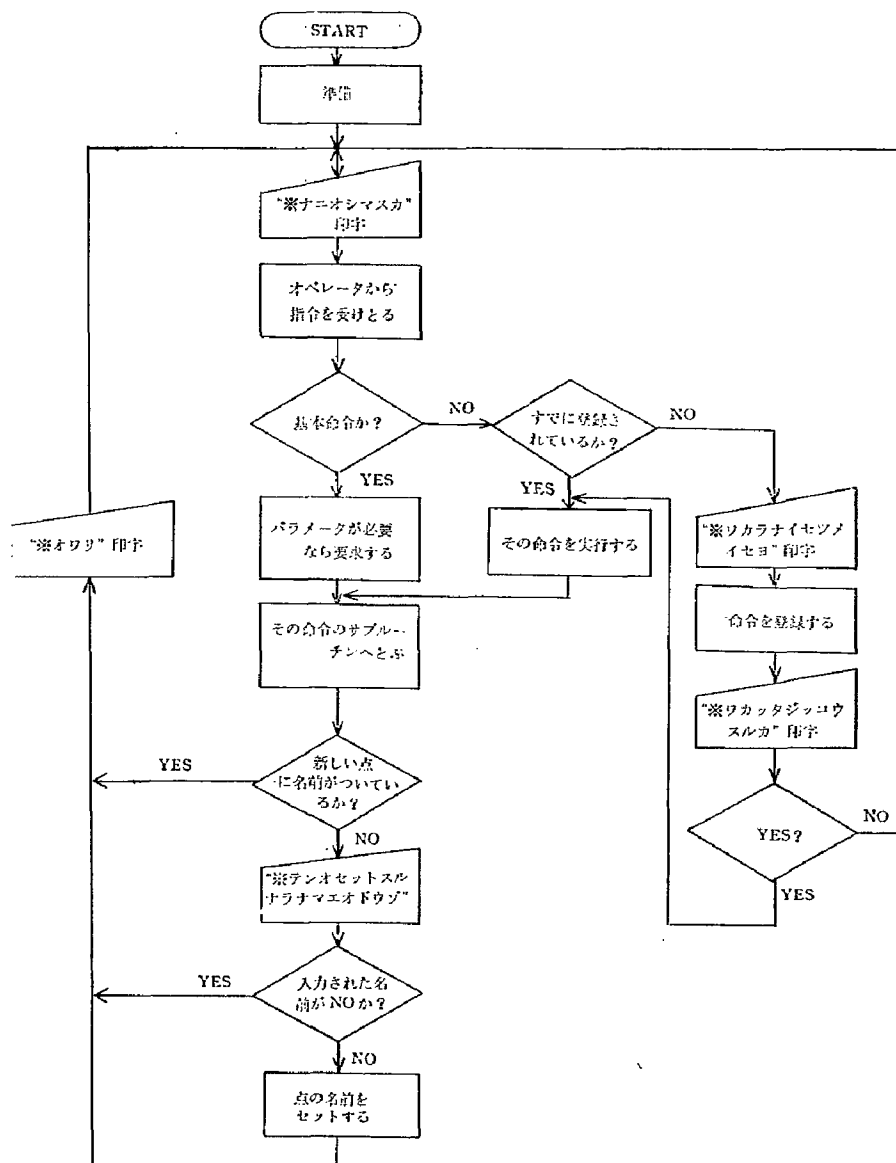


図 4: 主プログラムのフローチャート



がら各種の作業を教えて記憶させ、それまでに記憶している作業を自由に組み合わせて、更に新しい作業を構成していくためのプログラムの内容について述べる。

#### 4-1 主プログラムと情報の流れ

主プログラムにおける情報の流れの概略を図4に示す。この図に従って説明して行く。動く前の準備では、まず手首と腕のリセットを行なう。計算機のコアには、各軸の今までに出したパルスのカウントしているカウンタがソフトウェアとして設けられており、そのカウンタを参照することによって現在の位置を知ることができる。腕のリセットを行なうたびにカウンタも0にセットする。次に、磁気テープに前回までに登録した点や作業（命令）が入っているのを、それらをコアに読み込んで準備完了ということになる。オペレータからの指令を受けとるためにタイプライタ上に

※ナニ オ シマスカ

と印字し、オペレータからの動作指令を持つ。与えられた命令が基本命令であれば、この基本命令がパラメータを必要としなかったり、パラメータが与えられた場合はそのままそのサブルーチンに飛び、パラメータが与えられていなければ、パラメータを要求してから

飛ぶ。それらのパラメータは、点に名前をつけた記号名で与えられる場合も、データ（X軸、Y軸、Z軸の各値をmm単位で表わしたもの）で与えられる場合もある。データで与えられ、腕が動いたときは、その位置には一般に名前が付いていないと考えられるから

※テン オ セット スルナラ ナマエ オ ドウゾと印字してその位置に対する名前を要求する。登録したければその名前を入力すれば登録されるし、NOと入力すれば

※（作業名）オワリ

を印字し、始めに戻って次の動作指令を待つ。登録済みの点のみを使用した作業の場合は、動作完了後、直ちに※（作業名）オワリを印字して次の動作指令を待つ。

基本命令でなければ、磁気テープから入力した登録済みの命令の中にオペレータからの命令の名前があるかどうかを探し、あればその命令を実行する。なければ

※（作業名）ハ ワカラナイ セツメイ セヨ

を印字して命令群を要求する。それらの命令群の構成要素は基本命令でもよいし、前に登録した命令でもよい。与えられた命令群の中に、再び登録されていない

点領域	
GEN	
TEN	
	0
	0
	0
— Name 1 —	
	R1
	$\theta$ 1
	Z1
— Name 2 —	
	R2
	$\theta$ 2
	Z2
— Name 3 —	
	R3
	$\theta$ 3
	Z3
End mark	

図 5. 点領域の登録形式

命令領域	編集領域
Start mark	Start mark
— Name A —	— Name D —
パラメータ処理	⋮
命令コード	⋮
パラメータ	Start mark
End mark	— Name C —
Start mark	
— Name B —	End mark
パラメータ処理	
命令コード	
"CALL" mark	
Aの番地	
命令コード	
End mark	
Block mark	

図 6. 命令領域の登録形式

ものがあれば、また※（作業名）ハ ワカラナイ セツメイセヨ を印字して、それに対する命令群を要求する。これは何重になっていてもよい。命令を登録し終るとそれを実行するかどうかを聞く。NO と答えれば実行しないで始めに戻るし、YES と印字すれば実行して始めに戻る。仕事を終わりたい場合には、※ナニオ スマスカ に対して STOP と入力すれば、今までに登録した点と命令をすべて磁気テープに出力して計算機は PAUSE 状態で止まる。

#### 4-2. 点の登録、命令の登録および命令の実行

点の名前を登録するために1001語、命令（作業）を登録するために3000語の領域が確保されている。

まず、点の登録について述べる。点の座標はすべてパルス数に変換して記憶されている。一つの点を登録するために、点の名前に2語（6桁以内の記号）、R軸、 $\theta$  軸、Z軸の原点からのパルスの数に1語ずつ計5語

を必要とするから、このプログラムでは200点を登録できることになる。この点の名前だけを探索して行けば、入力された点の名前があるかいないかがわかり、また、それに対応するR、 $\theta$ 、Zの座標（RESET状態からのパルス数）もわかる。点の登録は、現在の位置を登録する場合はカウントの内容をそのまま座標とし、タイプライタより入力されたデータはパルス数に変換した後で終りの印のある位置（図5参照）から登録し、終りの印をその後につける。なお、初期設定で、GENTEN という名前の点を、座標（0, 0, 0）として自動的に登録する。

命令の登録は少しこみ入っている。命令の長さは一定ではないので、命令の登録領域では（図6参照）命令の名前の前に始めの印を入れ、命令群の終りには終りの印を記入する。命令を探すときには、始めの印を探して、その後の名前が探しているものと一致するか

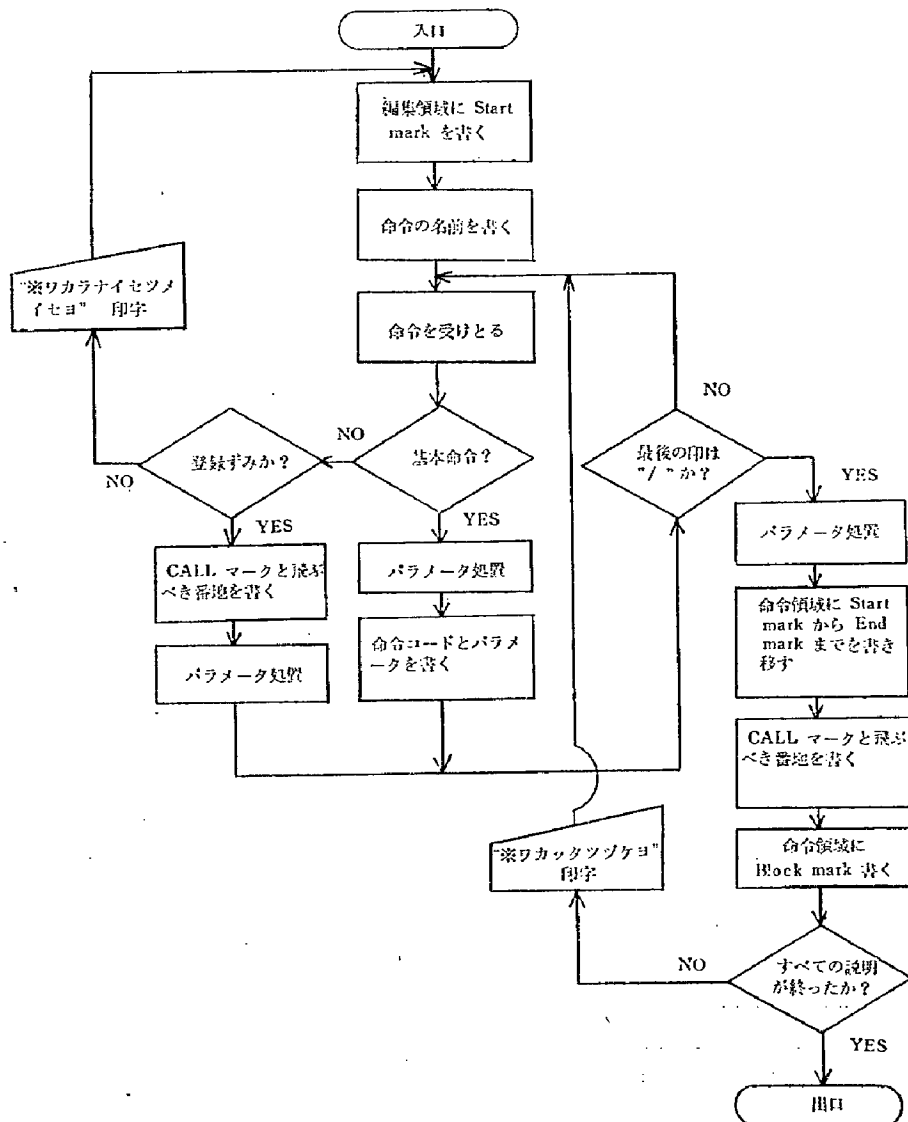


図 7. 命令登録のフローチャー

どうかを調べる。命令の登録のために、編集領域を設けて、タイプライタから入力する命令群が完結したら（／の印を入力したら）正式に命令の登録領域に移す。命令を登録するプログラムの FLOW CHART を図 7 に示す。まず、まだ登録されていない命令が与えられたときは、編集領域に始めの印をつけ、その次に与えられた命令の名前を書く。次に、可変なパラメータ（命令を実行する際にオペレータが入力できるパラメータ）をチェックするための番地を 1 語用意している。現在は 9 個まで可変なパラメータを持つことができる。オペレータに命令の説明を求めて、入力された命令が基本命令か登録済みの命令かによって次の処理が異なる。基本命令のときは基本命令のコードを記入し、もしパラメータが与えられればその次にそのデータを記入する。基本命令でないときは、命令登録の領域を探索し、もしあれば、CALL の記号を記入してから、その命令が登録されている番地を記入する。もし、その命令が登録されていないければ、再び始めの印を書き、同様に今迄の操作を繰り返す。／印が入力されると、終りの印を書き込んで、その前にある始めの印までが一つの命令として命令の登録領域に書き移され、編集領域には、CALL の印を記入して、次に今登録した命令の番地を書いて続ける。／の入力により検出した始めの印が、編集領域の最初にあったものならば、全部の説明が終ることになり、主プログラムへ戻る。登録された命令を実行するには、命令の登録領域を探索して、その命令があったら次に可変パラメータの番地を参照して、パラメータの処理をしながら以下に続く命令を実行する。すなわち、次に続く命令が基本命令のコードならば相当する基本命令を実行し、CALL の印があったら戻るべき番地をセットしてその次の番地の内容が示す番地に飛び、そこに記入されている命令群を実行して終りの印が来たら戻る。飛んだ命令群の中に再び CALL の印があれば再び戻るべき番地をセットして飛ぶ。このセットする場所に 50 語確保しているから、サブルーチンの中でサブルーチンを呼ぶ深さは、50 回まで許されることになる。

## § 5. あとがき

対話形式による工業用ロボットの計算機制御用ソフトウェアシステムについてその概要を説明した。以下に述べるように、今後の発展の方向を示唆するいろいろな問題を含んでいるが、人間とロボットとの協調という意味でのロボットソフトウェアシステムに関する基礎的な検討は本論文によって一応つくされていると思われる。

なお、本研究の特長は、

1) オペレータが実際にロボットの動きを監視しながら、計算機との対話によっていろいろな仕事を教え込むことができる。

2) ロボット自身が経験を積むことによってだんだん利口になり、オペレータの負担が漸次軽くなる

3) ロボットの智能だけで判断できないところは、オペレータが関与して適切な判断を下せる

4) 最初から、すべてを想定した作業プログラムを作る必要はなく、そのつど、ロボットが内蔵している仕事を適宜に組み合わせることにより自由にかつ簡単に作業命令が構成できる

などである。

さて、ロボットの機能を更に向上させるために解決せねばならぬ問題点としてあげることのできるものは、ほぼ、次の 4 点であろう。すなわち、

1) 各種センサ、特に眼からの情報に基づいて命令実行の流れを変えるブランチ命令を持たせること

2) 簡単な計算ルーチンを on-line で付加すること

3) 命令形式に柔軟性を与えること

4) 過去の経験を基にして新しい仕事をロボット自身が作り出していく能力を持たせること。

1), 2) の点については本質的な困難さはなく、しかもこの能力を付加することによってロボットの機能はかなり拡大させることができる。すなわち、これにより人間とロボットとの協調を密にでき、周囲の状況によってその手順を変更するなど適応能力をロボットに持たせることができる。この点については、本研究に続く研究により、ほぼ実現することができた。

また、ある程度のパラメータ学習により作業の熟練度を増して行くことも可能であろう。

3), 4) についてはかなりの高度な内容を含んでおり、世界的にみても、いわゆる智能ロボットとしてその基礎的研究がやっと結についたばかりであり、具体的な応用が可能となるのはかなりの年数を要するものと思われる。

これらの事柄は、一つには自然言語の機械言語への翻訳、あるいは、非常に柔軟性をもったロボット言語の開発といった問題を含んでいるし、他方には、problem solving や heuristic programming, learning program, associative memory など artificial intelligence, あるいは、semantic information processing といった未解決の情報処理の諸問題をも内蔵している。これらの諸問題については、今後の研究課題として取りくんで行こうと思う。

なお、本研究は、電子技術総合研究所で進められて

いる「智能をもった産業用ロボットに関する研究」の一環をなし、著者はその末端に参加させて頂いたもので、プロジェクトリーダーである上滝制御部長および野田電子計算機部長、佐藤システム制御研究室長、長田情報制御研究室長、述バイオニクス研究室長をはじめとする各研究室研究グループ各位の有益なご討論ご助言に対し深甚の謝意を表する。特に、長田情報制御研究室長には全面的に御指導を頂き、理科大実習生室井良一くん（現日本電気）にはプログラムの作成に関して御協力頂いた。ここに深く感謝致します。最後に、本学後藤以紀教授の不断の御指導に対して感謝致します。

## 文 献

- \*1 : H. A. Ernst : MH1-a computer operated mechanical hand, Doctoral Thesis, M. I. T., (1961)
- \*2 : W. R. Ferrell, T. B. Sheridan : Supervisory control of remote manipulation, IEEE spectrum, VOL 4, 81/88 (1967-10)
- \*3 : T. B. Sheridan, W. R. Ferrell : Human control of remote computer-manipulators, International Joint Conference on Artificial Intelligence, 483/487, (1969)
- \*4 : D. J. Barber : MANTRAN : A Symbolic Language for Supervisory Control of an Intelligent Remote Manipulator, M. S. Thesis, M. I. T. (1967)
- \*5 : 柿倉, 山口, 佐藤 : 触覚を有する マニピュレータの制御, 電四連大会, 2566(1970-4)
- \*6 : 上滝, 佐藤, 長田, 土屋, 白川, 向殿 : 工業用ロボットの計算機制御, 第1回バイオメカニズムシンポジウム (1970-8)
- \*7 : 長田, 向殿, 室井 : 対話形式による工業用ロボット計算機制御の一方式, 計測自動制御学会論文集, VOL 6, NO. 6(1970-12)